

## LISTING OF CLAIMS

1. (Currently Amended) A computer-implemented method for passing a message from a first thread of execution in a process to a second thread of execution in the process, comprising:

instantiating a first thread queue for holding messages, the first thread queue being associated with the first thread of execution ~~at the second thread of execution in the process, each queue comprising a reference to a further queue of the same type;~~

obtaining a first address associated with the first thread queue;

sending the first address to the second thread of execution, the second thread of execution being configured to create a second thread queue, to obtain a second address associated with the second thread queue, and to store the first address;

~~interpreting a block of source code at the first thread of execution in the process;~~

receiving the second address from the second thread in response to sending the first address;

~~instantiating~~ creating the message at the first thread of execution in the process;

obtaining a reference associated with the message;

placing, by the first thread of execution, a the reference to the message into the second thread queue of the second thread of execution, wherein the reference is usable by the second thread of execution to access the message.

2. (Canceled)

- 1 3. (Currently Amended) The computer-implemented method of claim 1,  
2 wherein placing the reference comprises placing the reference into the second  
3 thread queue based on the second address further comprising:  
4 receiving a reference to the second thread of execution's queue; and  
5 using the reference to the second thread of execution's queue to perform the  
6 placing step.
- 7 4. (Currently Amended) The computer-implemented method of claim 1,  
8 further comprising sending another message from the second thread to the first  
9 thread based on the first address wherein the first thread of execution has a  
10 queue, the method further comprising: passing, to the second thread of  
11 execution, a reference to the first thread of execution's queue to allow the  
12 second thread of execution to send messages to the first thread of execution.
- 13 5. (Currently Amended) The computer-implemented method of claim 1  
14 further comprising: sending a signal to the second thread of execution to  
15 indicate that a the message has been sent to the second thread of execution.
- 16 6. (Previously Presented) The computer-implemented method of claim 5,  
17 wherein the signal is sent via a platform-independent object.
- 18 7. (Currently Amended) The computer-implemented method of claim 1,  
19 wherein creating the message comprises further comprising:  
20 defining a message object for holding the message; and  
21 inserting the message into the message object, wherein the reference placed  
22 in the second thread of execution's queue is a reference to the message object.  
23  
24  
25

1 8. (Currently Amended) A method for passing intraprocess messages  
2 between scripting threads in a process, the method comprising:

3 creating a first scripting thread of execution;

4 creating a first thread queue for the first scripting thread, ~~the first queue~~  
5 ~~comprising a reference to a second queue of the same type as the first queue;~~

6 obtaining a first address associated with the first thread queue;

7 creating a second scripting thread of execution; and

8 passing, to the second scripting thread, the address ~~a reference to the first~~  
9 ~~scripting thread's queue~~ for use by the second scripting thread to send messages to  
10 the first scripting thread.

11 9. (Canceled)

12 10. (Currently Amended) The method of claim 8 further comprising:

13 creating a second thread queue for the second scripting thread;

14 obtaining a second address associated with the second thread queue; and

15 passing, to the first scripting thread, the second address ~~a reference to the~~  
16 ~~queue of the second scripting thread~~ for use by the first scripting thread to send  
17 messages to the second scripting thread.

18 11. (Currently Amended) The method of claim 8 further comprising:

19 creating a message object;

20 inserting a the message from the first scripting thread into the message  
21 object;

22 obtaining a reference to the message object; and

23 placing a the reference ~~to the message object~~ into the second thread queue  
24 ~~of the second scripting thread~~ so that the second scripting thread can access the  
25 message.

12. (Currently Amended) The method of claim 11 further comprising:

1 sending a signal from the first scripting thread to the second scripting  
2 thread to indicate to the second scripting thread that the ~~a new~~ message has been  
3 sent to the second scripting thread.

13. (Currently Amended) The method of claim 11, wherein in response to  
the message further comprising:

4 inserting a flag in the message object to indicate that ~~it~~ the message object  
5 is being responded to; and placing a reference to the message object into the first  
6 thread queue ~~of the first scripting thread~~.

1 14. (Currently Amended) A method for compiling a program having a  
2 plurality of sections, the method comprising:

3 creating, for each section of the program, a scripting thread that executes a  
4 script for compiling the section, wherein the script is independent of the program;  
5 and

6 creating a control thread to asynchronously communicate with each of the  
7 scripting threads so that commands can be issued from the control thread to the  
8 scripting threads in parallel, wherein asynchronously communicating with each of  
9 the scripting threads is via a plurality of message queues, each scripting thread  
10 being associated with one of the plurality of message queues and the control  
11 thread being associated with at least one of the plurality of message queues.

12 15. (Canceled)

13 16. (Previously Presented) The method of claim 14 further comprising:  
14 at the control thread, sending updates to a user interface; and  
15 processing commands from the user interface in parallel with  
16 asynchronously sending commands to the scripting threads.

17 17. (Currently Amended) The method of claim 14 further comprising:  
18 ~~creating a queue for the control thread; and~~  
19 obtaining a reference associated with the message queue associated with the  
20 control thread; and  
21 passing, to at least one of the scripting threads, a the reference ~~to the control~~  
22 ~~thread's queue~~ for use by the scripting thread to send messages to the control  
23 thread.  
24  
25

1 18. (Currently Amended) A system for compiling a program having a  
2 plurality of sections, the system comprising:

3 a computer;

4 a script for compiling each section of the program, wherein the script is  
5 independent of the program;

6 a plurality of scripting threads executing on the computer, wherein each  
7 section of the program is compiled under the direction of the script executed by a  
8 scripting thread of the plurality; and

9 a control thread executing on the computer for coordinating the activity of  
10 the scripting threads by communicating asynchronously with the scripting threads  
11 via a plurality of message queues.

12 19. (Currently Amended) The system of claim 18, wherein communicating  
13 asynchronously occurs via an interface of a cross-platform object further  
14 comprising: a means for allowing the control thread to communicate  
15 asynchronously with the scripting threads.

16 20. (Currently Amended) The system of claim 18 ~~further comprising: a~~  
17 ~~plurality of queues~~, wherein each message queue is associated with a scripting  
18 thread of the plurality of scripting threads, and wherein each message queue is  
19 adapted to receive messages from the control thread.

20 21. (Currently Amended) The system of claim 18 further comprising: a  
21 means for sending a signal from the control thread to at least one of the plurality  
22 of scripting threads to alert the scripting thread ~~whenever a~~ the message is sent to  
23 the scripting thread.  
24  
25

1 22. (Previously Presented) The system of claim 18 further comprising: a script  
2 engine executing on the computer, wherein the script engine interprets scripting  
3 language commands for each of the plurality of scripting threads and provides a  
4 means for sending a signal from the control thread to at least one of the plurality  
5 of scripting threads to alert the scripting thread whenever a message is sent to the  
6 scripting thread.

7 23. (Original) The system of claim 18, wherein the computer is a  
8 first computer, the system further comprising: at least one second computer in  
9 communication with the first computer, wherein at least one of the scripting  
10 threads executes on the second computer.

11 24. (Previously Presented) The system of claim 23 further comprising:  
12 a network link for enabling the first and second computers to communicate  
13 with one another;  
14 a means for allowing the scripting thread executing on the second computer  
15 to communicate across the network link with the control thread executing on the  
16 first computer.

17 25. (Previously Presented) The system of claim 18 further comprising: a user  
18 interface, wherein the control thread is operable to update the user interface  
19 without having to wait for the scripting threads to act on messages sent to them by  
20 the control thread.  
21  
22  
23  
24  
25

27. (Original) The system of claim 26, wherein the control thread sends messages asynchronously to each of the plurality of scripting threads to coordinate their activities, thereby resolving interdependencies among different sections of the program that are being compiled.



1 28. (Previously Presented) The system of claim 26, wherein the control  
2 thread is associated with one or more control thread queues out of the plurality  
3 of message queues and each scripting thread is associated with a scripting  
4 thread queue out of the plurality of message queues; further comprising:

5 ~~a one or more control thread queues associated with the control thread; and~~  
6 ~~a plurality of scripting thread queues, wherein each scripting thread queue~~  
7 ~~is associated with a scripting thread of the plurality of scripting threads, and~~  
8 ~~wherein the control thread has a reference to each scripting thread queue,~~  
9 and

10 ~~wherein each scripting thread has a reference to at least one control thread~~  
11 ~~queue that is associated with the scripting thread, thereby enabling the control~~  
12 ~~thread to put one or more of the messages in each scripting thread queue and each~~  
13 ~~scripting thread to put response messages in the associated queue of the control~~  
14 ~~thread.~~

15 29. (Previously Presented) The system of claim 26 further comprising:

16 at least one script stored on the server computer, wherein the script contains  
17 instructions for directing the compilation of the program; and

18 a script engine executing on the server computer to interpret the script, the  
19 script engine leaving an inter-thread signaling mechanism,

20 wherein the control thread uses signaling mechanism to alert a scripting  
21 thread of the plurality of scripting threads whenever the control thread has sent a  
22 message to the scripting thread.  
23  
24  
25

30. (Currently Amended) The system of claim 26, wherein the control thread sends messages asynchronously to each of the plurality of scripting threads to coordinate their activities, thereby resolving interdependencies among different sections of the program that are being compiled, wherein the control thread is associated with one or more control thread queues out of the plurality of message queues and each scripting thread is associated with a scripting thread queue out of the plurality of message queues;~~the system further comprising:~~

~~a plurality of control thread queues associated with the control thread;~~  
~~a plurality of scripting thread queues, wherein each scripting thread queue is associated with a scripting thread of the plurality of scripting threads, and~~

~~wherein~~ the control thread has a reference to each scripting thread queue, and

~~wherein~~ each scripting thread has a reference to a corresponding control thread queue of the plurality of control thread queues, thereby enabling the control thread to put one or more of the messages in each scripting thread queue and each scripting thread to put response messages in its corresponding control thread queue;

at least one script stored on the server computer, wherein the script contains instructions for directing the compilation of the program; and

a script engine executing on the server computer to interpret the script, the script engine having an inter-thread signaling mechanism, wherein the control thread uses signaling mechanism to alert a scripting thread of the plurality of scripting threads whenever the control thread has sent a message to the scripting thread.